

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-32. (Cancelled)

33. (Currently amended) A method for executing an application that is encapsulated in a package, the method comprising:

loading the package within a browser on a local client computer, the package including a manifest and a root archive of files that ~~include~~ includes instructions and content needed to execute the application, the root archive of files including:

an initial file that includes instructions for initiating execution of the application,

other files needed to execute the application, and

a nested archive ~~of that includes~~ additional files, the nested archive ~~of additional files being nested~~ included within the root archive of files ~~package~~;

the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the root archive of files when the application is executed;

in response to loading the package within the browser, automatically accessing the manifest;

locating the initial file identifier in the manifest;

based on the located initial file identifier, accessing the instructions for initiating the execution of the application ~~in~~ from the initial file;

processing the accessed instructions from the initial file; and

automatically initiating execution of the application based on the processed instructions.

34. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, further comprising receiving the package at the local client computer.

35. (Previously presented) The method for executing an application that is encapsulated in a package of claim 34, wherein receiving the package at a local client computer comprises loading the package onto the local client computer from a local computer readable medium.

36. (Currently amended) The method for executing an application that is encapsulated in a package of claim 33, wherein the instructions for initiating execution of the application ~~in~~ from the initial file include instructions to process additional instructions from one or more of the other files in the root archive, the method further comprising:

accessing the additional instructions ~~in~~ from the one or more other files; and
processing the additional instructions, the accessing and processing of the additional instructions being performed in response to processing the instructions for initiating execution of the application ~~in~~ from the initial file.

37. (Currently amended) The method for executing an application that is encapsulated in a package of claim 33, wherein:

the initial file comprises a source file for a web page,
the instructions for initiating execution of the application ~~in~~ from the initial file comprise instructions for rendering the web page, and
automatically initiating execution of the application comprises rendering the web page in accordance with the instructions for rendering the web page.

38. (Previously presented) The method for executing an application that is encapsulated in a package of claim 37, wherein the source file for the web page comprises an HTML document.

39. (Currently amended) The method for executing an application that is encapsulated in a package of claim 33, wherein:

the initial file comprises an executable file, and
the instructions for initiating execution of the application ~~in~~ from the initial file comprise program execution instructions.

40. (Previously presented) The method for executing an application that is encapsulated in a package of claim 39, wherein the executable file is a JAVAscript file.

41. (Currently amended) The method for executing an application that is encapsulated in a package of claim 39, wherein the instructions for initiating execution of the application ~~in~~ from the initial file comprise JAVA-based instructions.

42. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, wherein:

the manifest further comprises an archive type identifier that identifies an application type of the application, and

processing the accessed instructions comprises processing the accessed instructions in accordance with the application type of the application.

43. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, further comprising displaying an icon associated with the application, wherein loading the package in the browser comprises automatically loading the package in the browser in response to a user selecting the icon associated with the application.

44. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, wherein the initial file identifier indicates that the initial file is the first file to be processed in the root archive of files when the application is executed.

45. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, wherein automatically initiating execution of the application based on

the processed instructions comprises automatically initiating execution of the application without maintaining a connection between the local client computer and a web server.

46. (Currently amended) A computer program product for executing an application that is encapsulated in a package, the computer program product being embodied in a tangible computer readable medium and including instructions that, when executed by a processor, cause the processor to:

load the package within the computer program product, the package including a manifest and a root archive of files that include instructions and content needed to execute the application, the root archive of files including:

an initial file that includes instructions for initiating execution of the application,

other files needed to execute the application, and

a nested archive of that includes additional files, the nested archive of ~~additional files~~ being nested included within the ~~root archive of files package~~;

the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the root archive of files when the application is executed;

automatically access the manifest in response to loading the package within the computer program product;

locate the initial file identifier in the manifest;

access the instructions for initiating the execution of the application ~~in~~ from the initial file based on the located initial file identifier;

process the accessed instructions from the initial file; and

automatically initiate execution of the application based on the processed instructions.

47. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 46, further comprising instructions that, when executed by the processor, cause the processor to receive the package at a local client computer.

48. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 47, wherein the instructions that, when executed by the processor, cause the processor to receive the package at the local client computer comprise instructions that, when executed by the processor, cause the processor to load the package onto the local client computer from a local computer readable medium.

49. (Currently amended) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein the instructions for initiating the execution of the application ~~in~~ from the initial file include instructions to process additional instructions from one or more of the other files in the root archive, the computer program product further comprising instructions that, when executed by the processor, cause the processor to:

access and process the additional instructions ~~in~~ from the one or more other files in response to processing the instructions for initiating execution of the application ~~in~~ from the initial file.

50. (Currently amended) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein:

the initial file comprises a source file for a web page,

the instructions for initiating execution of the application ~~in~~ from the initial file comprise instructions for rendering the web page, and

the instructions that, when executed by the processor, cause the processor to automatically initiate execution of the application comprises instructions that, when executed by the processor, cause the processor to render the web page in accordance with the instructions for rendering the web page.

51. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 50 wherein the source file for the web page comprises an HTML document.

52. (Currently amended) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein:

the initial file comprises an executable file, and

the instructions for initiating execution of the application ~~in~~ from the initial file comprise program execution instructions.

53. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 52, wherein the executable file is a JAVAscript file.

54. (Currently amended) The computer program product for executing an application that is encapsulated in a package of claim 52, wherein the instructions for initiating execution of the application ~~in~~ from the initial file comprise JAVA-based instructions.

55. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 52, wherein:

the manifest further comprises an archive type identifier that identifies an application type of the application, and

the instructions that, when executed by the processor, cause the processor to process the accessed instructions comprise instructions that, when accessed by the processor, cause the processor to process the accessed instructions in accordance with the application type of the application.

56. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 46 further comprising instructions that, when executed by a processor, cause the processor to display an icon associated with the application, wherein the instructions that, when executed by the processor, cause the processor to load the package in the computer program product comprise instructions that, when executed by the processor, cause the processor to automatically load the package in the computer program product in response to a user selecting the icon associated with the application.

57. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein the computer program product comprises a browser.

58. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein the initial file identifier indicates that the initial file is the first file to be processed in the root archive of files when the application is executed.

59. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein the instructions that, when executed by the processor, cause the processor to automatically initiate execution of the application based on the processed instructions comprise instructions that, when executed by the processor, cause the processor to automatically initiate execution of the application without maintaining a connection between the local client computer and a web server.

60. (Currently amended) A method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer, the method comprising:

generating a root archive of files that include instructions and content needed to execute the application, the root archive including:

an initial file that includes instructions for initiating execution of the application,
other files needed to execute the application, and

a nested archive of that includes additional files, ~~the nested archive of additional files being nested within the root archive of files;~~

generating a manifest file that is associated with the root archive, the manifest file including an initial file identifier that instructs the browser to process the initial file before processing other files in the root archive in order to initiate execution of the ~~application-~~
application; and

encapsulating the root archive of files, including the nested archive, and the manifest file within a package.

61. (Currently amended) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 60, wherein the instructions for initiating execution of the application ~~in~~ from the initial file include instructions to process additional instructions from one or more of the other files in the root archive.

62. (Currently amended) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 60, wherein:

the initial file comprises a source file for a web page, and

the instructions for initiating execution of the application ~~in~~ from the initial file comprise instructions for rendering the web page.

63. (Previously presented) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 62, wherein the source file for the web page comprises an HTML document.

64. (Currently amended) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 60, wherein:

the initial file comprises an executable file, and

the instructions for initiating execution of the application ~~in~~ from the initial file comprise program execution instructions.

65. (Previously presented) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 64 wherein the executable file is a JAVAscript file.

66. (Currently amended) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 64, wherein the instructions for initiating execution of the application ~~in~~ from the initial file comprise JAVA-based instructions.

67. (Currently amended) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 60, wherein:

the manifest file further comprises an archive type identifier that instructs the browser to process the instructions for initiating execution of the application ~~in~~ from the initial file in accordance with the application type of the application.

68. (Currently amended) A self-contained package for distributing an application, the package being embodied in a tangible computer readable medium and configured to enable the application to be automatically executed within a browser environment, the package comprising:
a root archive including:

an initial content source having instructions for initiating execution of the application within the browser environment,

additional files that include instructions and content needed to execute the application within the browser environment, and

a nested archive ~~of that includes~~ other files, the nested archive ~~of other files~~ being ~~nested~~ included within the ~~root archive of files~~ self-contained package;

a manifest associated with the root archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the root archive when the application is executed.

69. (Previously presented) The self-contained package according to claim 68, wherein said initial content source is an HTML file containing content layout instructions for rendering a document that displays content included in the root archive.

70. (Previously presented) The self-contained package according to claim 68, wherein said initial content source is an executable file containing program execution instructions.

71. (Previously presented) The self-contained package according to claim 70, wherein said executable file is a JavaScript file.

72. (Previously presented) The self-contained package of claim 70, wherein the manifest further comprises an archive type identifier that identifies an application type of the application.

73. (Cancelled)

74. (Previously presented) The self-contained package of claim 68, wherein the additional files comprise one or more of a web page, a script, an image, a sound file, and a JAVA file.

75. (Currently amended) A web browser configured to receive and execute an application encapsulated in a self-contained package, the web browser comprising:

an input module for inputting the self-contained package, wherein the self-contained package includes:

a root archive including:

an initial content source having instructions for initiating execution of the application,

additional files that include instructions and content needed to execute the application,

a manifest associated with the root archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the root archive when the application is executed, and

a nested archive of that includes other files, the nested archive of other files being ~~nested~~ included within the root archive self-contained package;

a processing engine for:

accessing the manifest,

locating the initial file content identifier in the manifest,

based on the located initial file content identifier, accessing the instructions for initiating execution of the application ~~in~~ from the initial file content source, and

processing the instructions for initiating execution of the application ~~in~~ from the initial file content source before processing the instructions ~~in~~ from the additional files;
and

a content rendering and layout module for rendering content according to the instructions ~~in~~ from the initial content source and the additional files.

76. (Previously presented) The web browser according to claim 75, wherein said initial content source is an HTML file containing content layout instructions for rendering a document that displays content included in the root archive.

77. (Previously presented) The web browser according to claim 75, wherein said initial content source is an executable file containing program execution instructions.

78. (Previously presented) The web browser according to claim 77, wherein said executable file includes JAVAscript instructions.

79. (Previously presented) The web browser according to claim 77, wherein said executable file includes JAVA-based instructions.

80. (Previously presented) The web browser according to claim 75, wherein the input module is configured to input the self-contained software package from a computer readable medium.

81. (Previously presented) The web browser according to claim 80, wherein said computer readable medium is a compact disc containing digitally recorded information.

82. (Currently amended) The method for executing an application that is encapsulated in a package of claim 33, wherein:

~~the nested archive of additional files is the initial file and the nested archive of additional files includes~~ is included within another package, the other package being included within the package and including:

another initial file that includes instructions for ~~initiating execution of~~ executing the application, and

another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive ~~of files~~ when the application is executed; and

~~accessing the instructions for initiating the execution of the application includes:~~

the method further comprises:

accessing the other manifest after locating the initial file identifier in the manifest,
and

locating the other initial file identifier in the ~~manifest;~~ other manifest, and

~~accessing the instructions for initiating the execution of the application includes~~
accessing the instructions for ~~initiating execution of~~ executing the application in from the other initial file.

83. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, wherein the nested archive is configured according to a .JAR type file structure.

84. (Previously presented) The method for executing an application that is encapsulated in a package of claim 33, wherein the nested archive of additional files is one of the other files needed to execute the application.

85. (Currently amended) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein:

~~the nested archive of additional files is the initial file and the nested archive of additional files includes~~ is included within another package, the other package being included within the package and including:

another initial file that includes instructions for ~~initiating execution of~~ executing the application, and

another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive of files when the application is executed; and

~~the instructions that, when executed by a processor, cause the processor to access the instructions for initiating the execution of the application include instructions that, when executed by a processor, cause a processor to:~~

the computer program product for executing an application that is included in a package further comprising instructions that, when executed by the processor, cause the processor to:

access the other manifest after locating the initial file identifier in the manifest,
and

locate the other initial file identifier in the manifest; other manifest, and

~~the instructions that, when executed by a processor, cause a processor to access the instructions for initiating the execution of the application include instructions that, when executed by a processor, cause a processor to access the instructions for initiating execution of~~ executing the application in from the other initial file.

86. (Previously presented) The computer program product for executing an application that is encapsulated in a package of claim 46, wherein the nested archive is configured according to a .JAR type file structure.

87. (Currently amended) The method of claim 60 wherein the nested archive of ~~additional files is the initial file and the nested archive of additional files includes:~~ is included within another package, the other package being included within the package and including:

another initial file that includes instructions for ~~initiating execution of~~ executing the application, and

another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive ~~of files~~ when the application is executed.

88. (Previously presented) The method of claim 60 wherein the nested archive is configured according to a .JAR type file structure.

89. (Currently amended) The self-contained package according to claim 68 wherein the nested archive ~~of other files is the initial content source and the nested archive of additional files includes;~~ is included within another self-contained package, the other self-contained package being included within the self-contained package and including:

another initial file that includes instructions for ~~initiating execution of~~ executing the application, and

another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive ~~of files~~ when the application is executed.

90. (Previously presented) The self-contained package according to claim 68 wherein the nested archive is configured according to a .JAR type file structure.

91. (Currently amended) The web browser according to claim 75 wherein the nested archive ~~of other files is the initial content source and the nested archive of additional files includes;~~ is included within another self-contained package, the other self-contained package being included within the self-contained package and including:

another initial file that includes instructions for ~~initiating execution of~~ executing the application, and

another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive ~~of files~~ when the application is executed.

92. (Previously presented) The web browser according to claim 75 wherein the nested archive is configured according to a .JAR type file structure.

93. (New) The method for executing an application that is encapsulated in a package of claim 34, wherein receiving the package at the local client computer comprises receiving an e-mail that includes the package at the local client computer.

94. (New) The method for encapsulating an application in a package such that the application may be automatically executed by a browser on a client computer of claim 60 further comprising, establishing security credentials for the initial file, the other files needed to execute the application, and the nested archive using a single signature for the package.

95. (New) The method for executing an application that is encapsulated in a package of claim 82, wherein the package and the other package are executable .JAR files.

96. (New) A method for executing an application, the method comprising:
loading, within a processing environment on a local client computer, a first package that includes a first manifest and a first archive of files, wherein:

the first manifest includes a first initial file identifier that identifies a particular file from within the first archive of files as a first initial file that is to be processed before other files in the first archive of files when the application is executed, and

the first archive of files includes:

the first initial file, the first initial file including instructions for initiating execution of the application,

other files needed to execute the application, and

a second package that is included within the first package, the second package including:

a second manifest that includes a second initial file identifier that identifies a particular file from within the second archive of files as a

second initial file that is to be processed before other files in the second archive of files when the application is executed, and

a second archive of files that includes:

the second initial file, the second initial file including instructions for executing the application, and additional files needed to execute the application;

in response to loading the first package within the processing environment, automatically accessing the first manifest;

locating the first initial file identifier in the first manifest;

based on the located first initial file identifier, accessing the instructions for initiating execution of the application from the first initial file;

processing the accessed instructions from the first initial file; and

automatically initiating execution of the application based on the processed instructions from the first initial file.

97. (New) The method for executing an application of claim 96, wherein the first package and the second package are executable .JAR files.

98. (New) The method for executing an application of claim 96 further comprising:
accessing the second manifest after locating the first initial file identifier in the first;
locating the second initial file identifier in the second manifest;
based on the located second initial file identifier, accessing the instructions for executing the application from the second initial file;
processing the accessed instructions from the second initial file; and
executing the application based on the processed instructions from the second initial file.